

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-007364

(43)Date of publication of application : 12.01.1999

(51)Int.Cl.

G06F 3/12
B41J 29/38

(21)Application number : 09-158697

(71)Applicant : CANON INC

(22)Date of filing : 16.06.1997

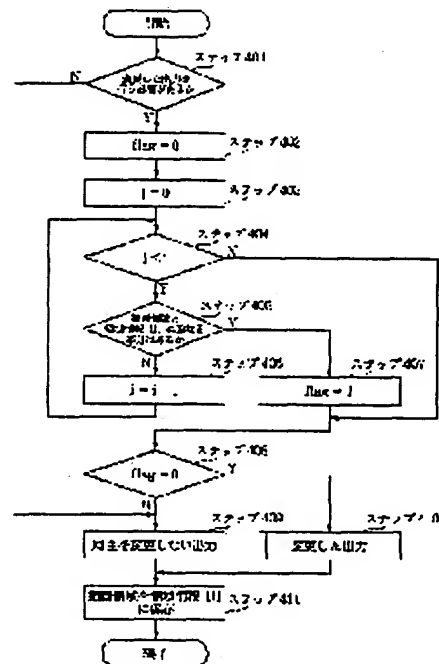
(72)Inventor : NAGOYA KENJI

(54) INFORMATION PROCESSOR AND METHOD THEREFOR AND STORAGE MEDIUM AND PRINTING SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To improve a processing speed, and to improve memory using efficiency by operating alteration related with a logical plotting output by managing a plotting area at the time of operating the plotting output from an inputted plotting request, improving a processing speed, and managing the plotting area with plural rectangles.

SOLUTION: At the time of inputting a plotting instruction, whether or not a plotting area of the inputted plotting instruction is overlapped on the plotting area of another plotting instructions is discriminated (S405). When it is discriminated that those plotting areas are overlapped, the plotting attribute of the inputted plotting instruction is not changed, and when it is discriminated that those areas are not overlapped, the plotting attribute of the inputted plotting instruction is changed (S409 and S410).



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-7364

(43) 公開日 平成11年(1999) 1月12日

(51) Int.Cl.⁶

識別記号

F I

G 0 6 F 3/12

G 0 6 F 3/12

C

B 4 1 J 29/38

B 4 1 J 29/38

Z

審査請求 未請求 請求項の数26 O L (全 13 頁)

(21) 出願番号 特願平9-158697

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(22) 出願日 平成9年(1997) 6月16日

(72) 発明者 名古屋 健二

東京都狛江市中和泉5-9-1

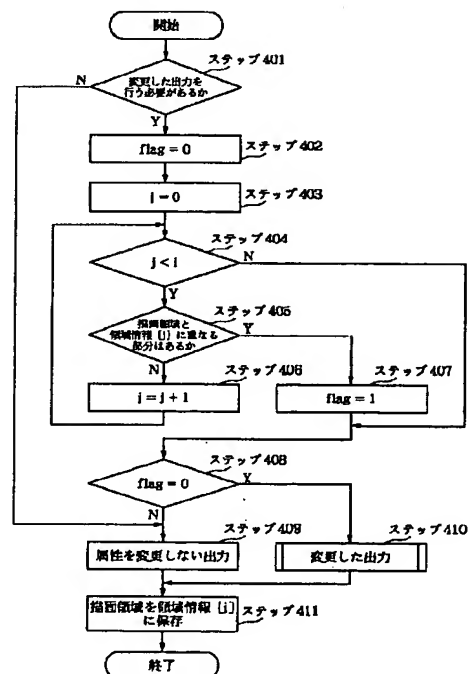
(74) 代理人 弁理士 丸島 儀一

(54) 【発明の名称】 情報処理装置、情報処理方法、記憶媒体、及び、印刷システム

(57) 【要約】

【課題】 入力された描画要求から描画出力を行う際に、描画領域の管理を行うことで論理描画出力に関する変更を行い、処理速度を向上させる。また、描画領域の管理を複数、矩形で行なうことにより、処理速度の向上およびメモリ使用効率の向上を図る。

【解決手段】 描画命令を入力する入力手段 (S 3 0 2) と、前記入力手段により入力された描画命令の描画領域が他の描画命令の描画領域と重なるか否かを判別する判別手段 (S 4 0 5) と、前記判別手段により重なりと判別された場合、前記入力手段により入力された描画命令の描画属性を変更せず、前記判別手段により重ならないと判別された場合、前記入力手段により入力された描画命令の描画属性を変更する変更手段 (S 4 0 9, S 4 1 0) とを有することを特徴とする。



【特許請求の範囲】

【請求項 1】 描画命令を入力する入力手段と、前記入力手段により入力された描画命令の描画領域が他の描画命令の描画領域と重なるか否かを判別する判別手段と、前記判別手段により重なりと判別された場合、前記入力手段により入力された描画命令の描画属性を変更せず、前記判別手段により重ならないと判別された場合、前記入力手段により入力された描画命令の描画属性を変更する変更手段とを有することを特徴とする情報処理装置。

【請求項 2】 前記変更手段により変更された描画命令をプリンタに出力する出力手段を有することを特徴とする請求項 1 記載の情報処理装置。

【請求項 3】 前記変更手段は、描画の種類の変更、または、描画の省略を行うことを特徴とする請求項 1 記載の情報処理装置。

【請求項 4】 前記変更手段は、OR 描画を上書き描画に変更し、AND 描画を省略し、XOR 描画を上書き描画に変更することを特徴とする請求項 1 記載の情報処理装置。

【請求項 5】 前記変更手段は、OR 描画を省略し、AND 描画を上書き描画に変更し、XOR 描画を上書き描画に変更し色属性を反転させることを特徴とする請求項 1 記載の情報処理装置。

【請求項 6】 前記入力手段により入力された描画命令が上書きのときは、前記判別手段による判別と前記変更手段による変更とを行わないことを特徴とする請求項 1 記載の情報処理装置。

【請求項 7】 前記判別手段による判別と前記変更手段による変更とを行うか否かを指定する指定手段を有することを特徴とする請求項 1 記載の情報処理装置。

【請求項 8】 描画領域を矩形ブロックに分割して前記判別手段による判別を行うことを特徴とする請求項 1 記載の情報処理装置。

【請求項 9】 描画命令を入力する入力ステップと、前記入力ステップにより入力された描画命令の描画領域が他の描画命令の描画領域と重なるか否かを判別する判別ステップと、前記判別ステップにより重なりと判別された場合、前記入力ステップにより入力された描画命令の描画属性を変更せず、前記判別ステップにより重ならないと判別された場合、前記入力ステップにより入力された描画命令の描画属性を変更する変更ステップとを有することを特徴とする情報処理方法。

【請求項 10】 前記変更ステップにより変更された描画命令をプリンタに出力する出力ステップを有することを特徴とする請求項 9 記載の情報処理方法。

【請求項 11】 前記変更ステップは、描画の種類の変更、または、描画の省略を行うことを特徴とする請求項 9 記載の情報処理方法。

【請求項 12】 前記変更ステップは、OR 描画を上書き描画に変更し、AND 描画を省略し、XOR 描画を上

書き描画に変更することを特徴とする請求項 9 記載の情報処理方法。

【請求項 13】 前記変更ステップは、OR 描画を省略し、AND 描画を上書き描画に変更し、XOR 描画を上書き描画に変更し色属性を反転させることを特徴とする請求項 9 記載の情報処理方法。

【請求項 14】 前記入力ステップにより入力された描画命令が上書きのときは、前記判別ステップによる判別と前記変更ステップによる変更とを行わないことを特徴とする請求項 9 記載の情報処理方法。

【請求項 15】 前記判別ステップによる判別と前記変更ステップによる変更とを行うか否かを指定する指定ステップを有することを特徴とする請求項 9 記載の情報処理方法。

【請求項 16】 描画領域を矩形ブロックに分割して前記判別ステップによる判別を行うことを特徴とする請求項 9 記載の情報処理方法。

【請求項 17】 描画命令を入力する入力ステップと、前記入力ステップにより入力された描画命令の描画領域が他の描画命令の描画領域と重なるか否かを判別する判別ステップと、前記判別ステップにより重なりと判別された場合、前記入力ステップにより入力された描画命令の描画属性を変更せず、前記判別ステップにより重ならないと判別された場合、前記入力ステップにより入力された描画命令の描画属性を変更する変更ステップと実行させるためのプログラムを記憶したコンピュータで読み出し可能な記憶媒体。

【請求項 18】 前記変更ステップにより変更された描画命令をプリンタに出力する出力ステップを有することを特徴とする請求項 17 記載の記憶媒体。

【請求項 19】 前記変更ステップは、描画の種類の変更、または、描画の省略を行うことを特徴とする請求項 17 記載の記憶媒体。

【請求項 20】 前記変更ステップは、OR 描画を上書き描画に変更し、AND 描画を省略し、XOR 描画を上書き描画に変更することを特徴とする請求項 17 記載の記憶媒体。

【請求項 21】 前記変更ステップは、OR 描画を省略し、AND 描画を上書き描画に変更し、XOR 描画を上書き描画に変更し色属性を反転させることを特徴とする請求項 17 記載の記憶媒体。

【請求項 22】 前記入力ステップにより入力された描画命令が上書きのときは、前記判別ステップによる判別と前記変更ステップによる変更とを行わないことを特徴とする請求項 17 記載の記憶媒体。

【請求項 23】 前記判別ステップによる判別と前記変更ステップによる変更とを行うか否かを指定する指定ステップを有することを特徴とする請求項 17 記載の記憶媒体。

【請求項 24】 描画領域を分割して前記判別ステップ

による判別を行うことを特徴とする請求項 1 7 記載の記憶媒体。

【請求項 2 5】 前記プログラムは、プリンタドライバプログラムであることを特徴とする請求項 1 7 記載の記憶媒体。

【請求項 2 6】 プリンタと、情報処理装置とを有する印刷システムにおいて、前記情報処理装置は、描画命令を入力する入力手段と、前記入力手段により入力された描画命令の描画領域が他の描画命令の描画領域と重なるか否かを判別する判別手段と、前記判別手段により重なりと判別された場合、前記入力手段により入力された描画命令の描画属性を変更せず、前記判別手段により重ならないと判別された場合、前記入力手段により入力された描画命令の描画属性を変更する変更手段と、前記変更手段により変更された描画命令をプリンタに出力する出力手段とを有することを特徴とする印刷システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は情報処理装置、情報処理方法、コンピュータで読み出し可能なプログラムを記録した媒体、および、印刷システムに関するものである。

【0002】

【従来の技術】従来の描画出力生成では、OR描画やAND描画といった該当する領域の状態に依存する論理描画出力処理を、該当する領域の内容によらず処理しており、印字不良の発生、論理描画属性の切り替えに伴う出力データ量の増大、プリンタ等の出力装置での印刷スループットの低下、プリンタ等の出力装置での再現性の低下といった欠点があった。

【0003】

【発明が解決しようとする課題】上述の通り、従来の描画出力生成では、OR描画やAND描画といった該当する領域の状態に依存する論理描画出力処理を、該当する領域の内容によらず処理しており、印字不良の発生、論理描画属性の切り替えに伴う出力データ量の増大、プリンタ等の出力装置での印刷スループットの低下、プリンタ等の出力装置での再現性の低下といった欠点があった。

【0004】

【課題を解決するための手段】上記課題を解決するため本発明は、従来実現できなかった一部の論理描画が可能となることによる印字不良発生の低下、論理描画属性の切り替えに伴う出力データ量の肥大化の防止、プリンタ等の出力装置での印刷スループットの向上、プリンタ等の出力装置での再現性の向上を目的とする。

【0005】また、描画要求回数が多かった場合の処理速度を向上させることを目的とする。

【0006】また、処理を簡素化させ、処理速度を向上させることを目的とする。

【0007】また、記憶を効率的に行い、処理速度を向

上させることを目的とする。

【0008】

【課題を解決するための手段】上記目的を達成するために本発明の情報処理装置は、描画命令を入力する入力手段と、前記入力手段により入力された描画命令の描画領域が他の描画命令の描画領域と重なるか否かを判別する判別手段と、前記判別手段により重なりと判別された場合、前記入力手段により入力された描画命令の描画属性を変更せず、前記判別手段により重ならないと判別された場合、前記入力手段により入力された描画命令の描画属性を変更する変更手段とを有することを特徴とする。

【0009】また、本発明の情報処理方法は、描画命令を入力する入力ステップと、前記入力ステップにより入力された描画命令の描画領域が他の描画命令の描画領域と重なるか否かを判別する判別ステップと、前記判別ステップにより入力された描画命令の描画属性を変更せず、前記判別ステップにより重ならないと判別された場合、前記入力ステップにより入力された描画命令の描画属性を変更する変更ステップとを有することを特徴とする。

【0010】また、本発明の記憶媒体は、描画命令を入力する入力ステップと、前記入力ステップにより入力された描画命令の描画領域が他の描画命令の描画領域と重なるか否かを判別する判別ステップと、前記判別ステップにより入力された描画命令の描画属性を変更せず、前記判別ステップにより重ならないと判別された場合、前記入力ステップにより入力された描画命令の描画属性を変更する変更ステップと実行させるためのプログラムを記憶する。

【0011】また、本発明の印刷システムは、プリンタと、情報処理装置とを有する印刷システムにおいて、前記情報処理装置は、描画命令を入力する入力手段と、前記入力手段により入力された描画命令の描画領域が他の描画命令の描画領域と重なるか否かを判別する判別手段と、前記判別手段により重なりと判別された場合、前記入力手段により入力された描画命令の描画属性を変更せず、前記判別手段により重ならないと判別された場合、前記入力手段により入力された描画命令の描画属性を変更する変更手段と、前記変更手段により変更された描画命令をプリンタに出力する出力手段とを有することを特徴とする。

【0012】

【発明の実施の形態】第 1 図は本発明の一発明の実施の形態を示す論理描画自動処理装置の構成を説明するブロック図である。なお、本発明の機能が実行されるのであれば、単体の機器であっても、複数の機器からなるシステムであっても、LAN、WAN等のネットワークを介して接続が為され処理が行われるシステムであっても本発明を適用できることは言うまでもない。

5

【0013】図において、3000はホストコンピュータで、ROM3のプログラム用ROMに記憶された文書処理プログラム等に基づいて図形、イメージ、文字、表（表計算等を含む）等が混在した文書処理を実行するCPU1を備え、システムデバイス4に接続される各デバイスをCPU1が総括的に制御する。

【0014】また、このROM3のプログラム用ROMあるいは外部メモリ11には、CPU1の制御プログラムであるオペレーティングシステムプログラム（以下OS）等を記憶し、ROM3のフォント用ROMあるいは外部メモリ11には上記文書処理の際に使用するフォントデータ等を記憶し、ROM3のデータ用ROMあるいは外部メモリ11には上記文書処理等を行う際に使用する各種データ（例えば、ディレクトリ情報、プリンタドライバテーブル等）を記憶する。この場合、制御の主体は、ハードウェア上はCPUである。一方、ソフトウェア上は制御の主体は本論理描画自動処理プログラムを含む印刷関連モジュールである。2はRAMで、CPU1の主メモリ、ワークエリア等として機能する。5はキーボードコントローラ（KBC）で、キーボード9や不図示のポインティングデバイスからのキー入力を制御する。6はCRTコントローラ（CRT）で、CRTディスプレイ（CRT）10の表示を制御する。7はディスクコントローラ（DKC）で、ブートプログラム、各種のアプリケーション、フォントデータ、ユーザファイル、編集ファイル、論理描画自動処理プログラム等を記憶するハードディスク12（HD）、フロッピーディスクドライブ13（FDD）にセットされたフロッピーディスク14（FD）等の外部メモリ11とのアクセスを制御する。8はプリンタコントローラ（PRTC）で、所定の双方向性インターフェース（インターフェース）20を介してプリンタ5000に接続されて、プリンタ5000との通信制御処理を実行する。なお、CPU1は、例えばRAM2上に設定された表示情報RAMへのアウトラインフォントの展開（ラスターライズ）処理を実行し、CRT10上でのWYSIWYGを可能としている。また、CPU1は、CRT10上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウインドウを開き、種々のデータ処理を実行する。また、描画命令変更する、しないを設定するウインドウを表示させ、ユーザに選択させる。

【0015】プリンタ5000において、21はプリンタCPUで、ROM23のプログラム用ROMに記憶された制御プログラム等あるいは外部メモリ30に記憶された制御プログラム等に基づいてシステムバス24に接続される印刷部（プリンタエンジン）27に出力情報としての画像信号を出力する。また、このROM23のプログラムROMには、CPU21の制御プログラム等を記憶する。ROM23のフォント用ROMには上記出力情報を生成する際に使用するフォントデータ等を記憶

6

し、ROM23のデータ用ROMにはハードディスク等の外部メモリ30がないプリンタの場合には、ホストコンピュータ上で利用される情報等を記憶している。CPU21は入力部25を介してホストコンピュータとの通信処理が可能となっており、プリンタ内の情報等をホストコンピュータ3000に通知可能に構成されている。22はCPU21の主メモリ、ワークエリア等として機能するRAMで、図示しない増設ポートに接続されるオプションRAMによりメモリ容量を拡張することができるように構成されている。なお、RAM22は、出力情報展開領域、環境データ格納領域、NVRAM等に用いられる。前述したハードディスク（HD）、ICカード等の外部メモリ30は、ディスクコントローラ（DKC）29によりアクセスを制御される。外部メモリ30は、オプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。また、28は前述した操作パネルで操作のためのスイッチおよびLED表示器等が配されている。

【0016】また、前述した外部メモリは1個に限らず、少なくとも1個以上備え、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラムを格納した外部メモリを複数接続できるように構成されていてもよい。さらに、図示しないNVRAMを有し、操作パネル28からのプリンタモード設定情報を記憶するようにしてもよい。

【0017】図2はホストコンピュータ3000において、OSの管理の基にRAM2にロードされた本制御モジュールが実行可能となった状態のメモリマップを示している。

【0018】図3は本発明の実施の形態の概略をフローチャートで示したものである。

【0019】図4は前記図3の出力データ生成ステップ（ステップ201）の詳細な処理をフローチャートで示したものである。

【0020】図5は前記図4の領域処理および出力処理ステップ（ステップ306）の詳細な処理をフローチャートで示したものである。

【0021】図6は前記図5のステップ410の詳細な処理をフローチャートで示したものである。

【0022】図7は前記図5の「描画領域を領域情報[i]に保存するステップ」（ステップ411）で保存する領域情報のデータを示したものである。

【0023】図3、4、5、6によって表わされるプログラムはホストコンピュータ3000上のHD12またはFD14に対して論理描画自動処理プログラムを含むモジュールとして記憶され、CPU1がOS46の管理の下で実行することにより達成する手段として機能している。

【0024】以下、図3、4、5、6に示すフローチャートを中心にして、本発明の発明の実施の形態を詳しく

説明する。

【0025】本発明の実施の形態では、ホストコンピュータ3000において、BIOS、OS、アプリケーション及び本発明における論理描画自動処理制御プログラムをCPU1が実行することにより動作する。BIOSはプログラムROM3に書き込まれており、OSは外部メモリ11であるハードディスク12（以下、HD12）に書き込まれている。そして、ホストコンピュータ3000の電源がONされた時に、BIOSプログラム中のIPL（Initial Program Load ing）機能により、OSがHDからRAM2へ読み込まれ、OSの動作が開始される。

【0026】そして、実際に、プリンタ色処理モード自動制御プログラムを含む印刷関連モジュールが動作可能となるのは、ユーザ等の指示により、ホストコンピュータ3000上で、OS管理の下動作するアプリケーションより印刷処理が実行された時で、印刷関連モジュールを記録しているFD14をFDD13にセットするか、または、印刷関連モジュールをHDドライブ12に保存しておき、OSおよびBIOSの制御のもとに印刷関連モジュールがFD14またはHDドライブ12から読み出され、RAM2にロードされた時である。前述の通り、第2図が、本発明の実施の形態における属性自動処理制御プログラムを含む印刷関連モジュールがホストコンピュータ3000上のRAM2にロードされ実行可能となった状態のメモリマップを示している。大まかな処理フローは図3に示す通りで印刷実行時にアプリケーションより受け取った印刷データから出力データ生成を行うステップ（ステップ201）からなっている。詳しくは図4に示す通り、まず保存した領域の数を示す変数*i*を0にする（ステップ301）。次にアプリケーションからの描画処理を受け付け（ステップ302）、それが描画終了の要求かどうかの判別を行う（ステップ303）。受け付けた描画処理が描画終了の場合、処理を終了する。受け付けた描画処理が描画終了ではない場合、その描画処理が改ページの要求であるかどうかの判別を行う（ステップ304）。その描画処理が改ページの要求である場合、変数*i*を0にし（ステップ305）、ステップ302に戻る。それ以外の場合、領域の管理および出力を行う（ステップ306）。次に変数*i*に1を加え（ステップ307）、次の描画処理の受け付け処理（ステップ302）に戻る。以降ステップ302からステップ307までの一連の処理をアプリケーションから描画終了の要求を受けるまで続ける。ステップ306の処理の詳細は図5に示す通りで、まず変更した出力を行う必要があるかどうかを判別する（ステップ401）。ここで、必要がないと判断されるのは、描画属性が上書きであり変更の必要がない場合、及び、ユーザにより描

画属性を変更しないモードが選択されている場合である。必要がない場合、ステップ409に進む。必要がある場合、変数flagを0にする（ステップ402）。次に変数*j*を0にする（ステップ403）。次に*j*が*i*より小さいかどうかの判別を行う（ステップ404）。*j*が*i*より小さくない場合、ステップ408に進む。*j*が*i*より小さい場合、アプリケーションより受けた描画要求の描画領域と保存した領域情報の*j*番目（以降、領域情報〔*j*〕と表記）に重なる部分があるかどうかを判別する（ステップ405）。重なる部分がある場合、変数flagを1にし（ステップ407）、ステップ408に進む。重なる部分がない場合、変数*j*に1を加え、ステップ404に戻る。以降ステップ404からステップ406の処理が繰り返され、ステップ404またはステップ405の条件によりステップ408に進む。ステップ408では変数flagが0であるかを判別する。0である場合、アプリケーションより受け付けた描画要求の領域が今までに描画したものと重なっていないため、該当領域は白である。そのため、アプリケーションが要求した論理描画を該当領域が白である場合に限定した同等処理に置き換えることができる。この変更した出力を行うのがステップ410である。次にステップ411に進む。ステップ410の詳細な処理は図6に示す通りで、描画を省略できるかどうかを判別する（ステップ451）。省略できる場合、描画を省略し（ステップ452）終了する。省略できない場合、属性を変更した出力を行う（ステップ453）。

【0027】属性を変更した場合の例を以下に示す。
濃度で表現する場合の例

- ・白を0、黒を1の2値で表現
- ・白を0、黒を255の256階調で表現 等

【0028】

【表1】

元の描画属性	変更した描画属性
OR 描画	上書き描画
AND 描画	描画の省略
XOR	上書き描画

【0029】輝度で表現する場合の例

- ・白を1、黒を0の2値で表現
- ・白を255、黒を0の256階調で表現
- ・RGBの順で赤を（255, 0, 0）緑を（0, 255, 0）青を（0, 0, 255）白（255, 255, 255）黒（0, 0, 0）で表現 等

【0030】

【表2】

9

10

元の描画属性	変更した描画属性	変更した色属性
OR 描画	描画の必要なし	変更なし
AND 描画	上書き描画	変更なし
XOR	上書き	元の色を反転したもの（元の色と白のXORをとったもの）

【0031】Windowsにおけるラスタオペレーションコードの例

【0032】

【表3】

元の描画属性	変更した描画属性
DSo	D（描画の必要なし）
DSa	S
DSx	DSn
DPa	P

【0033】なおDはディスティネーション、Sはソース、Pはパターン、oはOR描画、aはAND描画、xはXOR、nはNOTを表す。ちなみに、DSoは、ディスティネーションとソースをOR描画するためのコードで、それは、D（描画の必要なし）という描画属性に置き換えられる。

【0034】変数flagが0でない場合、該当領域には以前に描画されている。そのため、最終的な描画結果を変えないように、変更しない出力を行う（ステップ409）。次にステップ411に進む。ステップ411ではアプリケーションから受けた描画要求の描画領域を保存する。この際の領域情報の保存されかたを示したのが図7である。

【0035】今までに示した本発明の実施の形態により、従来実現できなかった一部の論理描画が可能となることによる印字不良発生の低下、論理描画属性の切り替えに伴う出力データ量の肥大化の防止、プリンタ等の出力装置での印刷スループットの向上、プリンタ等の出力装置での再現性の向上といった効果を実現する。

【0036】なお、本発明の実施の形態では本論理描画出力処理プログラムを含む印刷関連モジュールを記録する媒体をFDもしくはHDドライブとしたが、媒体はそれ以外にCD-ROMやICメモ리카ード、あるいは電子メールやパソコン通信等のネットワークを介するものであっても良い。更に論理描画出力処理プログラムを含む印刷関連モジュールをROM3に記録しておき、これをメモリマップの一部となすように構成し、直接CPU1で実行することも可能である。

【0037】（他の発明の実施の形態）また、前発明の実施の形態では領域の記憶を単一に行っていたが、その記憶を複数に分割して行うのが本発明の実施の形態で

言及するところである。

【0038】前発明の実施の形態で示した図4、5の処理を本発明の実施の形態に基づいて変更したのが図9、10である。

【0039】また、本発明の実施の形態では領域の記憶を複数で行うが、その記憶の分割を領域で行う際の参考に図8を示す。図8は1ページ分の領域を縦方向にa等分、横方向にb等分し、その結果c（=a×b）の領域に分割された様子を示している。本発明の実施の形態ではこの分割された各領域に対してそれぞれ描画領域を記憶する例を示す。

【0040】図9は描画領域の記憶を複数に分割した場合の、前記図3の出力データ生成ステップ（ステップ201）の詳細な処理をフローチャートで示したものである。

【0041】図10は前記図9の領域処理および出力処理ステップ（ステップ506）の詳細な処理をフローチャートで示したものである。

【0042】図11は前記図8の分割された各領域に対応する情報を保存するためのデータ構造である。（以下、分割領域情報）この中には対応する分割された各領域を保存するための分割領域、その領域内に描画したオブジェクトの数を示す保存オブジェクト数、そして今まで描画したオブジェクトの各領域が入っている。

【0043】図12は前記図11で示した分割領域情報が分割した領域の数（=c）だけ保存されているのを示す図である。このi番目の分割領域情報を、分割領域情報[i]と表記する。

【0044】図13はc個のそれぞれの分割領域情報の分割領域にどの領域が入っているかの対応を示した図である。

【0045】図14は描画の一例であり、本発明の実施の形態での説明に用いる。

【0046】図15は前記図8のa=1、b=2の場合の例である。

【0047】以下、図3、9、10に示すフローチャートを中心にして発明の実施の形態を詳しく説明する。

【0048】本発明の実施の形態は前記図3のステップ201が図9の場合である。まず、c個の分割領域情報の保存オブジェクト数を0にする（ステップ501）。次に図13に示すように、c個の分割領域情報の分割領域をセットする（ステップ502）。次にアプリケーションからの描画処理を受け付け（ステップ503）、描

画処理が描画終了の要求であるかを判別する（ステップ 504）。受け付けた描画処理が描画終了である場合、処理を終了する。受け付けた描画処理が描画終了でない場合、その描画処理が改ページの要求であるかどうかを判別する（ステップ 505）。改ページの要求であった場合、ステップ 501 と同様に c 個の分割領域情報の保存オブジェクト数を 0 にする（ステップ 507）。改ページの要求ではなかった場合、領域の管理および出力を行う（ステップ 506）。次に描画処理の受け付け処理（ステップ 503）に戻る。以降ステップ 503 から 507 までの一連の処理をアプリケーションから描画終了の要求を受けるまで続ける。ステップ 506 の処理の詳細は図 10 に示す通りで、まず変数 k を 0 にする（ステップ 601）。次に k が分割した領域の数 c より小さいかどうかの判定をおこなう（ステップ 602）。小さい場合は、終了する。小さい場合は次にステップ 603 に進む。ステップ 603 では描画領域と分割領域情報 $[k]$ の分割領域に重なる部分があるかを判別する。重なる部分がない場合、ステップ 616 に進む。重なる部分がある場合、変更した出力を行う必要があるかどうかを判別する（ステップ 604）。変更した出力を行う必要がない場合、ステップ 613 に進む。変更した出力を行う必要がある場合、変数 $flag$ を 0 にする（ステップ 605）。次に変数 j を 0 にする（ステップ 606）。次に j と分割領域情報 $[k]$ の保存オブジェクト数を比較する（ステップ 607）。 j の方が小さい場合、ステップ 611 に進む。 j の方が小さい場合、ステップ 608 に進む。ステップ 608 では描画領域と分割領域情報 $[k]$ の j 番目のオブジェクトの領域が重なるかどうかを判別する。重ならない場合、 j に 1 を加え（ステップ 609）、ステップ 607 に戻る。重なる場合、ステップ 610 に進み、変数 $flag$ を 1 にする。次に変数 $flag$ が 0 かどうかを判別する（ステップ 611）。変数 $flag$ が 0 の場合、変更した出力を行う（ステップ 612）。ステップ 612 の詳細な処理は前記図 6 の処理である。次にステップ 614 に進む。変数 $flag$ が 0 でない場合、変更しない出力を行い（ステップ 613）、ステップ 614 に進む。ステップ 614 では分割領域情報の保存オブジェクト数 (m と表わす) に 1 を加える。次に描画領域を分割領域情報 $[k]$ の m 番目のオブジェクトの描画領域として保存する（ステップ 615）。次にステップ 616 に進み、 k に 1 を加え、ステップ 602 に戻る。以降、ステップ 602 からステップ 616 の一連の処理をステップ 602 の条件で終了するまで行う。

【0049】これまでに示した発明の実施の形態について図 14 の具体例を用いて、説明を補足する。

【0050】ここで 2 つの領域が重なっているかどうかの判別には、他の処理に比べ、非常に時間のかかるものとする。また領域の分割は図 15 のように行うとする。

【0051】領域の記憶を 1 つにする場合、2 つの領域

が重なっているかどうかの判別（ステップ 405）は、 $2n$ 個目の描画オブジェクトの場合、最大 $(2n-1)$ 回必要である。

【0052】描画の記憶を 2 つにする場合、2 つの領域が重なっているかどうかの判別（ステップ 603 およびステップ 608）は $2+(n-1)$ 回必要である。

【0053】出力を行うまでには、
総判別回数 = 1 個目のオブジェクトの判別の回数 + 2 個目のオブジェクトの判別の回数 + ... + $2n$ 個目のオブジェクトの判別の回数

であるから、 n が大きい場合には領域の記憶を 1 よりも 2 にした方が回数が少なくて済む。

【0054】今回の例の場合、処理時間は判別の回数に依存するため、領域の記憶を複数 (= 2) に分割した方がはるかに高速に処理が行える。

【0055】本発明の実施の形態の場合、ページごとに用紙サイズが変わらない場合を示しているが、変わる場合でも同様である。

【0056】今まで示した発明の実施の形態により、描画要求回数が多くなった場合の処理速度を向上させることができるようになる。

【0057】また他の発明の実施の形態として前記 2 項目の発明の実施の形態で図 7 の領域の情報または図 11 のオブジェクトの描画領域として保存していた領域の情報を矩形で表わすというのが本発明の実施の形態で言及するところである。

【0058】これは図 16 のような描画の領域を保存する際に、それより小さいことがないような矩形の情報として保存するということであり、その一例を図 17 に示す。矩形以外の形態で保存する場合、図 16 の図形の各頂点の座標の形態、もしくは対応するピクセルが領域内であれば 1 そうでなければ 0 といったようなイメージに類する形態で保存する必要がある。この場合、描画領域が大きくて複雑な形状をしていると記憶領域を大きく消費する。また 2 つの領域が重なっている場合の判定にも時間を要する。

【0059】今まで説明したように、領域の保存形態を図 17 で示したような矩形で行うことにより、記憶を効率的に行い、処理速度を向上を図ることができる。

【0060】

【発明の効果】以上説明したように、本発明によれば、従来実現できなかった一部の論理描画が可能となることによる印字不良発生の低下、論理描画属性の切り替えに伴う出力データ量の肥大化の防止、プリンタ等の出力装置での印刷スループットの向上、プリンタ等の出力装置での再現性の向上といった効果がある。

【0061】また、描画要求回数が多くなった場合の処理速度を向上させるといった効果がある。

【0062】また、描画要求回数が多くなった場合の処理速度を向上させるといった効果がある。

13

【0063】また、記憶を効率的に行い、処理速度を向上させるといった効果がある。

【図面の簡単な説明】

【図1】本説明の一発明の実施の形態を示す情報処理装置の構成を示すブロック図である。

【図2】本発明の実施の形態における論理描画自動処理プログラムを含む印刷関連モジュールが情報処理装置3000上のRAM2にロードされ実行可能となった状態のメモリマップである。

【図3】本発明の実施の形態の概略を示したフローチャートである。

【図4】図3のステップ201の詳細な処理を示したフローチャートである。

【図5】図4のステップ306の詳細な処理を示したフローチャートである。

【図6】図5のステップ410の詳細な処理を示したフローチャートである。

【図7】図5のステップ411で保存する領域情報を示す図である。

【図8】描画領域の記憶を複数で行う際の分割を領域ご

とに行う場合の例を示す例である。

【図9】図3のステップ201の詳細な処理を示したフローチャートである。

【図10】図9のステップ506の詳細な処理を示したフローチャートである。

【図11】図9のステップ614で保存する分割領域情報の内容を示す図である。

【図12】図10で示した分割領域情報が分割した数の分保存されるのを示す図である。

【図13】図9のステップ502でそれぞれの分割領域情報の分割領域にどの領域が入っているかの対応を示す図である。

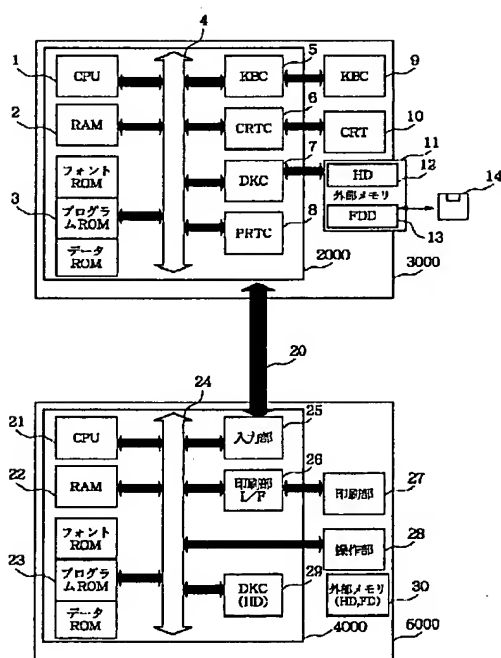
【図14】例における描画を示す図である。

【図15】描画領域の記憶を複数で行う際の分割の例を示す図である。前記図8のa=1, b=2の場合に相当する。

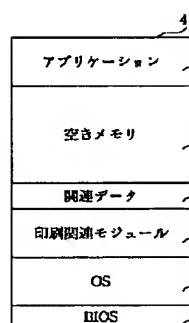
【図16】描画の一例を示す図である。

【図17】前記図16での描画の領域の保存を矩形で行う例を示す図である。

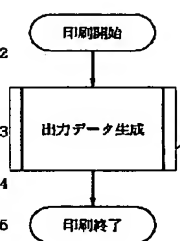
【図1】



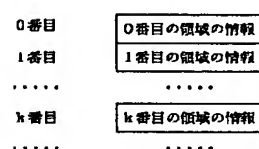
【図2】



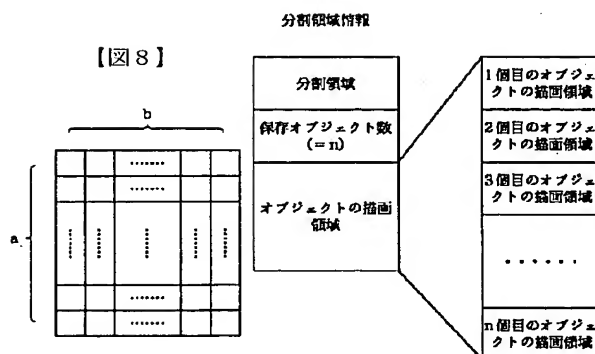
【図3】



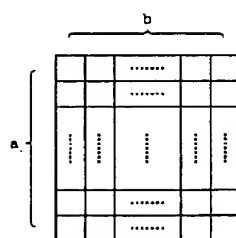
【図7】



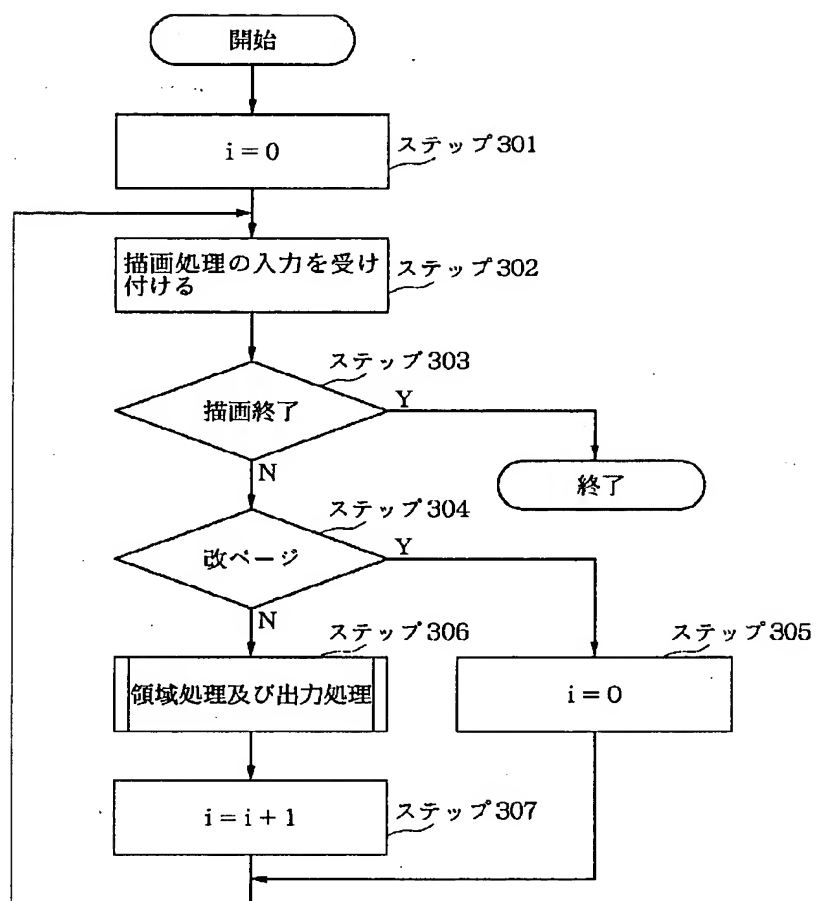
【図11】



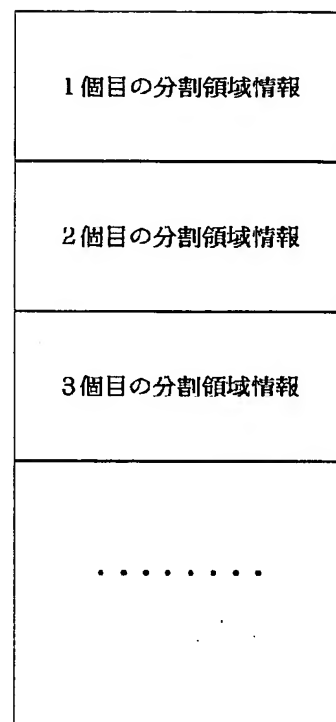
【図8】



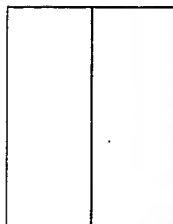
【図 4】



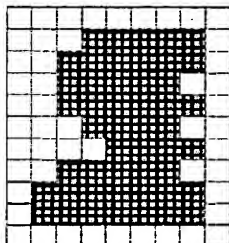
【図 1 2】



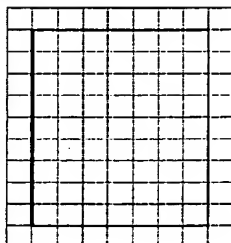
【図 1 5】



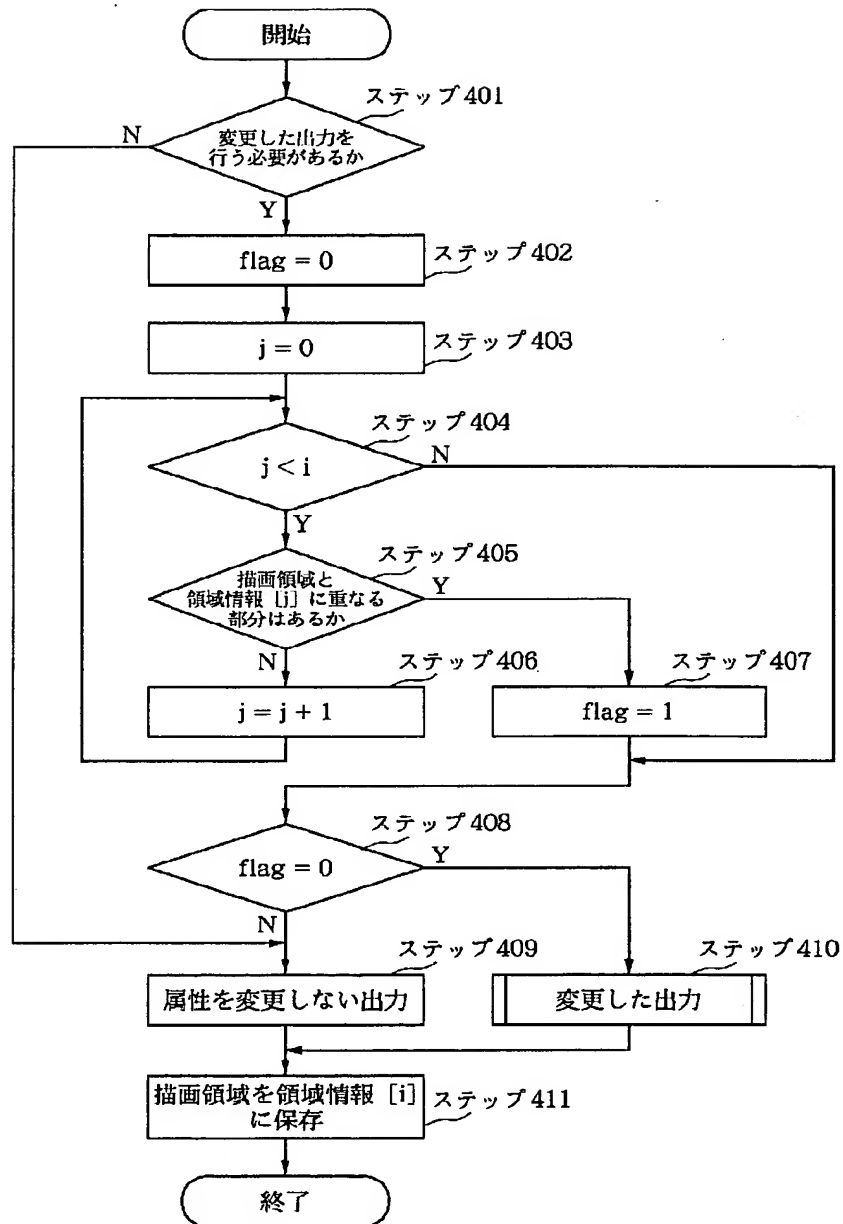
【図 1 6】



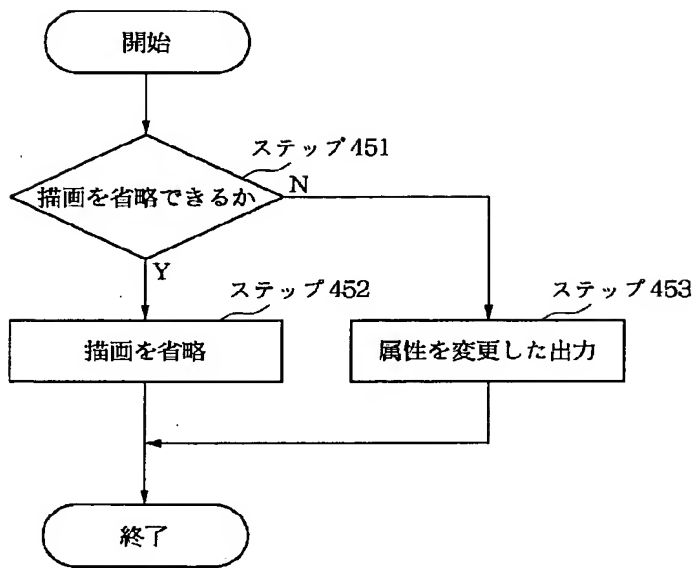
【図 1 7】



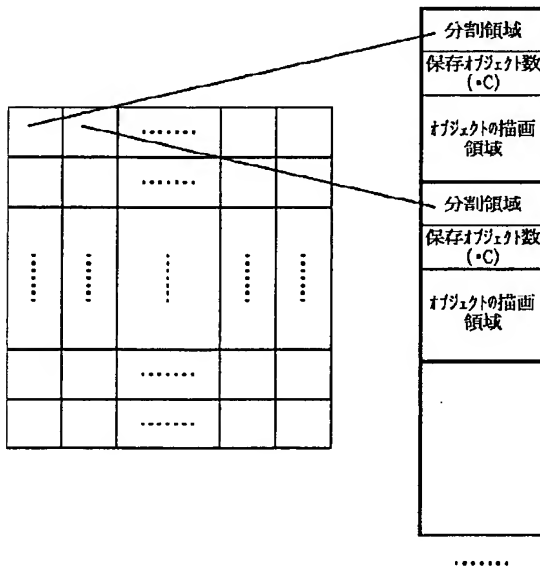
【図 5】



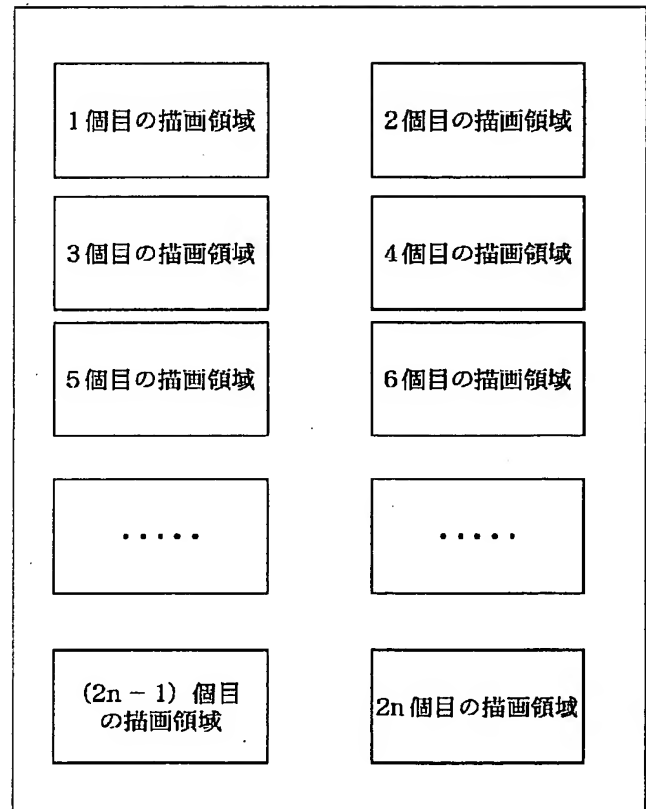
【図 6】



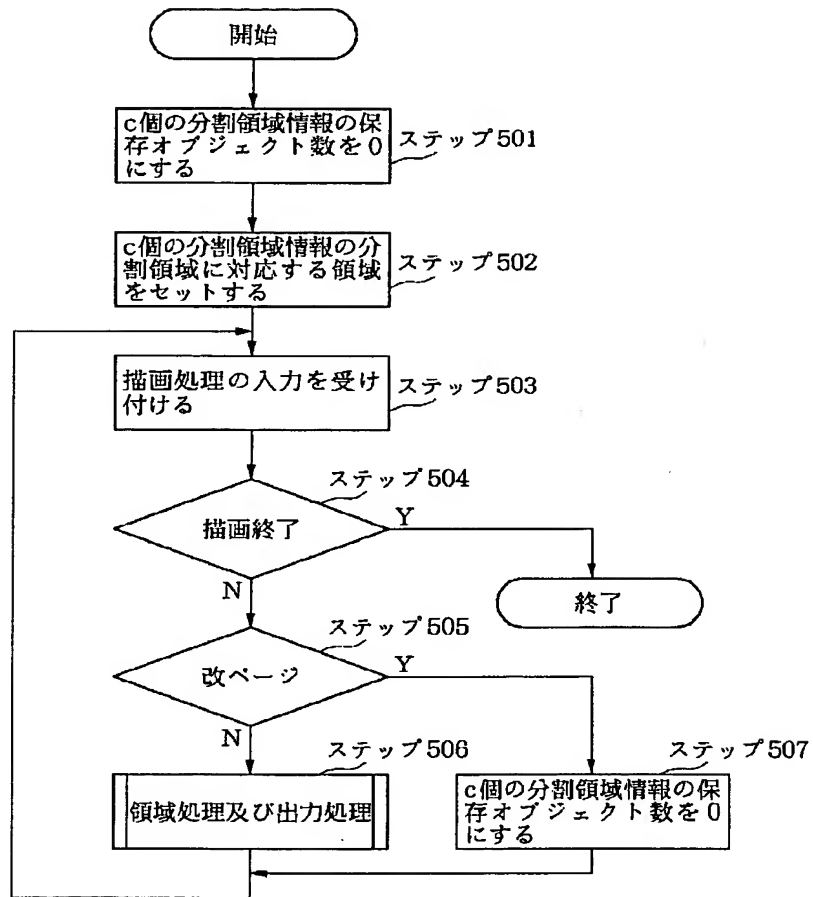
【図 1 3】



【図 1 4】



【図 9】



【図 10】

